The `gcdlike` function is similar to the Euclidean algorithm for computing the greatest common divisor (GCD) of two integers `p` and `q`. However, it has a slightly different purpose. Here is the function for reference:

```
public static boolean gcdlike(int p, int q) {
    if (q == 0) return (p == 1);
    return gcdlike(q, p % q);
}
```

Understanding the Function

This function checks whether the GCD of two integers `p` and `q` is `1`. In other words, it checks if `p` and `q` are **coprime** (i.e., they have no common divisors other than `1`). Let's break down the logic:

1. **Base Case**:
    - `if (q == 0) return (p == 1);`
    - When `q` becomes `0`, the function returns whether `p` is `1`. If `p` is `1`, it returns `true`; otherwise, it returns `false`.
2. **Recursive Case**:
    - `return gcdlike(q, p % q);`
    - The function recursively calls itself with `q` and `p % q`, continuing the Euclidean algorithm process.

How It Works

The Euclidean algorithm repeatedly replaces the larger number by its remainder when divided by the smaller number, until one of the numbers becomes `0`. The last non-zero remainder is the GCD of the original two numbers. In this function:

- If `p % q` is calculated and the function continues with the new pair `(q, p % q)`.
- Eventually, `q` becomes `0`, and the function checks if the current value of `p` is `1`.

Effect of the Function

The function `gcdlike(p, q)` returns `true` if and only if the GCD of `p` and `q` is `1`, meaning `p` and `q` are coprime. Here’s what it means in terms of integers and divisors:

1. **When `gcdlike(p, q)` Returns `true`**:
    - `p` and `q` have no common divisors other than `1`.
    - There are no integers greater than `1` that divide both `p` and `q`.
2. **When `gcdlike(p, q)` Returns `false`**:

- `p` and `q` have a common divisor greater than `1`.
- There exists at least one integer greater than `1` that divides both `p` and `q`.

Example Walkthrough

Let's consider an example with `p = 14` and `q = 9`:

1. `gcdlike(14, 9)`:
    - `q` is not `0`, so call `gcdlike(9, 14 % 9)` which is `gcdlike(9, 5)`.
2. `gcdlike(9, 5)`:
    - `q` is not `0`, so call `gcdlike(5, 9 % 5)` which is `gcdlike(5, 4)`.
3. `gcdlike(5, 4)`:
    - `q` is not `0`, so call `gcdlike(4, 5 % 4)` which is `gcdlike(4, 1)`.
4. `gcdlike(4, 1)`:
    - `q` is not `0`, so call `gcdlike(1, 4 % 1)` which is `gcdlike(1, 0)`.
5. `gcdlike(1, 0)`:
    - `q` is `0`, so check if `p` is `1`. Since `p` is `1`, return `true`.

Thus, `gcdlike(14, 9)` returns `true`, indicating that 14 and 9 are coprime.

Summary

The `gcdlike` function effectively determines whether two integers `p` and `q` are coprime. If the function returns `true`, the integers share no common divisors other than `1`, meaning they are coprime. If the function returns `false`, they share at least one common divisor greater than `1`.